



Arda EDEN¹

**SES SENTEZLEME TEKNİKLERİ ÖĞRETİMİNDE EĞİTİM
MATERYALİ OLARAK “PURE DATA”**

Özet

Ses sentezleme, çeşitli tekniklerden faydalanılarak yeni tınılar üretme süreci olarak tanımlanabilir. Ürünü ses olan bu sürece yönelik tekniklerin öğretilmesinde uygulama, tınısal sonuçların gözlemlenmesi ve nispeten soyut sayılabilecek teorik bilginin somutlaştırılması açısından önemlidir. Bu uygulamalar donanım veya yazılım tabanlı ses sentezleyiciler üzerinde gerçekleştirilebilir. Çalışmada, açık kaynaklı bir ses sentezleme ortamı olan Pure Data'nın ses sentezleme teknikleri öğretimi sürecinde eğitim materyali olarak kullanılabilirliği; yazılımın sunduğu görsellik ve gerçek zamanlılık özellikleri çerçevesinde, örnek uygulamalar ile desteklenerek tartışılmıştır.

Anahtar Kelimeler: Müzik teknolojisi eğitimi, Ses sentezleme, Pure Data.

**“PURE DATA” AS A TEACHING MATERIAL IN SOUND SYNTHESIS
TECHNIQUES EDUCATION**

Abstract

Sound synthesis can be defined as the process of creating new timbres by using various techniques. Implementation is so important during teaching the techniques used within the process, where the final product is sound, in terms of observing the timbral results and objectifying the relatively abstract theoretical information. The techniques can be implemented on both hardware and software synthesizers. In this study; the usage of Pure Data, an open source sound synthesis platform, as a teaching material in sound synthesis techniques education is discussed within the scope of visuality and real-time processing, supported by sample implementations.

Keywords: Music technology education, Sound synthesis, Pure Data.

¹ Yrd. Doç. Dr., İnönü Üniversitesi, Müzik Teknolojisi, ardaeden@gmail.com

GİRİŞ

Müziğin tarihsel süreç içerisindeki evrimine paralel olarak ortaya çıkan yeni tını arayışları, hızla değişen teknolojinin de sunduğu imkanlar ile birlikte ses sentezleme tekniklerine yönelik oldukça önemli bir bilgi birikiminin oluşmasına öncülük etmiştir. Ses sentezleme sürecinin ürünü sestir. O halde bu tekniklerin öğretimi sürecinde teorik bilginin aktarılmasının yanısıra, ilgili tekniklerin ses sentezleme araçları üzerinde uygulanarak sonuçların sınıf ortamında işitsel olarak sunulması da konunun anlaşılabilirliğinin artırılması açısından oldukça büyük önem taşımaktadır. Bu amaçla donanımsal veya yazılımsal ses sentezleyici araçlardan faydalanılabilir. Görsel arayüze sahip açık kaynak kodlu yazılımsal bir ses sentezleme ortamı olan Pure Data; görselliğinin sağladığı tasarım kolaylığıyla, ücretsiz olmasıyla ve gerçek zamanlı çıktı üretebilme özellikleriyle ses sentezleme teknikleri öğretimi sürecinde bir eğitim materyali olarak değerlendirilebilir.

Ses Sentezleme

Sentezleme; birleştirme, bir araya getirme, bir bütünü parçalarından oluşturma süreci olarak tanımlanır. Ses sentezleme ise ses üretme sürecidir. Bu süreçte hazır (var olan) seslerin çeşitli teknikler kullanılarak başkalaştırılması veya mekanik - elektronik araçlar ile özgün sesler elde etme gibi çeşitli yöntemlerden faydalanılabilir (Russ, 2004: 3-4).

Temel ses sentezleme yöntemleri analog ve sayısal yöntemler olarak iki başlık halinde sınıflandırılabilir. Sayısal yöntemler çoğunlukla, analog yöntemlerin sayısal teknolojiye uyarlanmış biçimi olarak ortaya çıkmaktadır (Russ, 2004: 7-10).

Temel ses sentezleme yöntemleri şu şekilde sıralanabilir:

- Artırmalı sentezleme (Additive synthesis)
- Eksiltmeli sentezleme (Subtractive synthesis)
- Genlik modülasyonu (Amplitude modulation – AM)
- Frekans modülasyonu (Frequency modulation – FM)
- Dalga şekillendirme (Waveshaping)
- Dalga tablosu (Wavetable)
- Fiziksel modelleme (Physical modelling)

Ses sentezleme işleminin gerçekleştirildiği araçlara sentezleyici (synthesizer) adı verilmektedir. Sentezleyicileri genel olarak iki ana grupta değerlendirmek mümkündür:

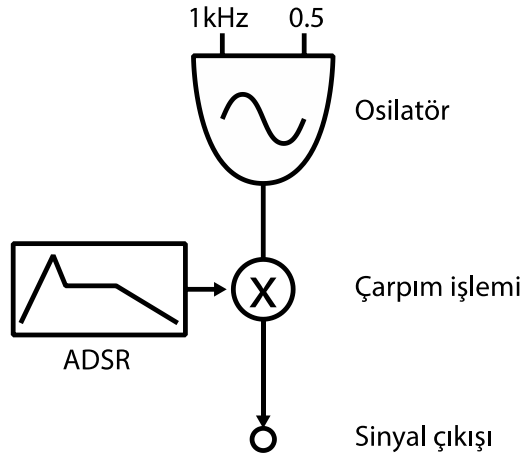
- Sabit mimari (hazır bağlantılı)
- Modüler (esnek bağlanabilir)

Günümüzde kullanılmakta olan sentezleyicilerin büyük bir çoğunluğu hazır bağlantılı olan sabit mimari ile üretilmiştir. Bu mimari sentezleyicinin taşınabilirliği ve pratikliği açısından avantajlıdır, ancak iç bağlantılar kullanıcı tarafından değiştirilemediğinden esnek yapıda değildir. Modüler sentezleyiciler ise bileşenleri ayrı modüller halinde üretilmiş araçlardır. Bu modüller arasındaki bağlantılar kullanıcı tarafından, ilgili sentezleme tekniğinin gerektirdiği biçimde yapılabilmektedir. Ses üretme teknikleri açısından oldukça geniş olanaklar sağlayan bu araçlar, eğitim ve araştırmaya yönelik akademik uygulamalarda da geniş kullanım alanı bulmaktadırlar (Önen, 2011: 37-38). Buna karşılık bu sentezleyiciler oldukça maliyetli araçlardır.

Grafiksel Notasyon

Ses sentezleme sürecinde kullanılan araçların ve iç bağlantıların görselleştirilmesi amacıyla grafik notasyondan faydalanılmaktadır. Grafik notasyon ilk olarak Music 4BF ve Music V gibi ilk modüler sayısal ses sentezleyici yazılımların işleyiş biçimlerini açıklamak amacıyla kullanılmıştır ve günümüzde halen geçerliliğini koruyan bir yöntemdir (Roads, 1996: 96).

Bu yöntemde her araç kendisine özgü bir şekil ile temsil edilir. Bu araçlar arasındaki bağlantılar ise sinyal akışını gösteren çizgiler veya oklar ile gösterilir. Grafiksel notasyon gerek uygulamada, gerekse sentezleme tekniklerin öğretilmesi sürecinde sinyal akışının daha somut bir biçimde ifade edilebilmesini sağlaması açısından oldukça kullanışlı bir yöntemdir (Şekil 1).



Şekil 1 – Basit bir sinüs osilatörünün genliğinin ADSR zarfı ile şekillendirilmesini gösteren grafiksel notasyon

Bilgisayar Tabanlı Ses Sentezleme

Bilgisayarlar ile ses sentezlemeye yönelik ilk çalışmalar 1957’de Bell Telefon Laboratuvarları’nda başlamıştır. İlk denemeler sonucunda Max V. Mathews ve arkadaşları bir bilgisayarın zaman karşı değişen frekans ve genlik zarfları ile birlikte tüm perde dizilerinde çeşitli dalga biçimlerindeki sesleri sentezleyebileceğini göstermişlerdir. Bu süreçte geliştirilen ilk ses sentezleyici yazılımlar Music I (1957) ve Music II (1958) olmuştur. 1960 yılında geliştirilen Music III ise UG (Unit Generator – Birim Üreteç) kavramının ilk kez ortaya çıktığı ve kullanıldığı yazılımdır. Birim üreteçler birbirleri ile iç bağlantılar üzerinden haberleşebilen, osilatörler, filtreler ve amplifikatörler gibi sinyal işleme araçlarıdır. Bu teknoloji sayesinde ses sinyallerinin bir dizi birim üreteç içerisinden geçirilerek çeşitli sentezleme algoritmalarının kolaylıkla uygulanabilmesi mümkün hale gelmiştir. İlerleyen süreçte Music III’ün yeni makro assembly dili ile yazılmış sürümü olan Music IV ve dönemin standart programlama dillerinden Fortran IV ile yazılmış Music V (1968) geliştirilmiştir. Music IV ve Music V temel alınarak geliştirilmiş olan sonraki nesil Music 4BF, Music 360, Music 7, Music 11, Csound, MUS10,

Cmusic ve Common Lisp Music ise “Music N” serisi yazılımlar olarak tarihteki yerlerini almışlardır (Roads, 1996, 89-90).

Kod Tabanlı Ses Sentezleme Yazılımları

Bilgisayarlar çeşitli donanım ve yazılımların bir araya gelmesiyle oluşan araçlardır. İlgili yazılım (programlar) olmadan bir bilgisayarın belirli bir işi yerine getirmesi mümkün değildir. Yazılımlar programcılar tarafından çeşitli programlama dilleri ile yazılarak oluşturulur. Bir programlama dili temel olarak bilgisayarın yerine getirmesi gereken talimatlar dizisi olarak tanımlanabilir. Bu talimatlar dizisi programcılar tarafından kaynak kodu olarak adlandırılır. Bir bilgisayarın doğrudan anlayabileceği koda makine kodu denir. Makine kodu ile programlama son derece karmaşık ve zor bir iştir. Bu nedenle programcının daha rahat anlayabileceği ve yazabileceği yapılarda farklı programlama dilleri geliştirilmiştir. Bu diller ile yazılan kaynak kodu ilgili derleyici veya yorumlayıcı program aracılığı ile bilgisayarın anlayacağı makine diline dönüştürülür. Farklı uygulamaların (yazılımların) geliştirilmesine uygun özelliklerde pek çok programlama dili bulunmaktadır. Music N serisi yazılımlar da ses sentezlemeye yönelik geliştirilmiş uygulamalar olmakla birlikte, aslında birer programlama dilidir.

Şekil 2(a)’da frekansı 440 Hz, genliği 1 olan sinüs dalganın grafiksel notasyonu görülmektedir. Şekil 2(b) ve Şekil 2(c)’de ise bu dalganın Csound ve Supercollider ortamlarında oluşturulabilmesi için gerekli kod örnekleri görülmektedir.



(a)

Csound

```
//Orkestra dosyası
instr 1
a1 oscil 1, 440, 1
out a1
endin

//Score dosyası
f 1 0 16384 10 1
i 1 0 5
```

(b)

Supercollider

```
{ SinOsc.ar(440, 0, 1) }.play;
```

(c)

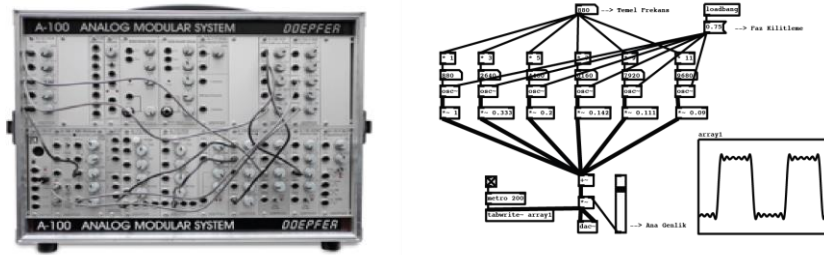
Şekil 2 - Basit bir sinüs osilatörün grafiksel notasyonu ile Csound ve Super Collider uygulaması

Csound ve Supercollider Linux, Windows ve Mac Os X platformlarında çalışabilen açık kaynak kodlu, ücretsiz olarak temin edilebilen yazılımlardır.

Pure Data

Pure Data ses, video ve grafiksel sinyalleri gerçek zamanlı (real-time) olarak işleyebilen grafiksel (görsel) bir programlama ortamıdır. Miller Puckette tarafından IRCAM’de geliştirilmiştir. Mac Os X, Linux ve Windows platformları üzerinde çalışabilen Pure Data açık kaynak kodlu özgür bir yazılımdır ve pek çok özgür yazılım gibi ücretsiz olarak edinilebilmektedir. Pure Data, Csound ve Supercollider gibi metin (kod) tabanlı programlama ortamlarından farklı olarak, sunduğu görsel programlama arayüzü ile programcıya büyük ölçüde kolaylık sağlamaktadır.

Pure Data’da fonksiyonlar (nesneler), operatörler ve tüm görsel kullanıcı arayüzleri eski modüler analog ses sentezleyicilere benzer şekilde birbirlerine kablolar aracılığı ile bağlanırlar. Bu sanal kablolar programlama arayüzünde çizgiler biçiminde gösterilir. Nesneler arasındaki veri (mesaj) iletimi bu sanal kablolar üzerinden gerçekleştirilir. Çeşitli donanımların (Pure Data’da nesnelerin) birbirlerine bu şekilde bağlanması “patching” olarak ifade edilir (Şekil 3). Bu nedenle Pure Data dosyaları “patch” olarak da adlandırılır (Chung, 2013: 8).

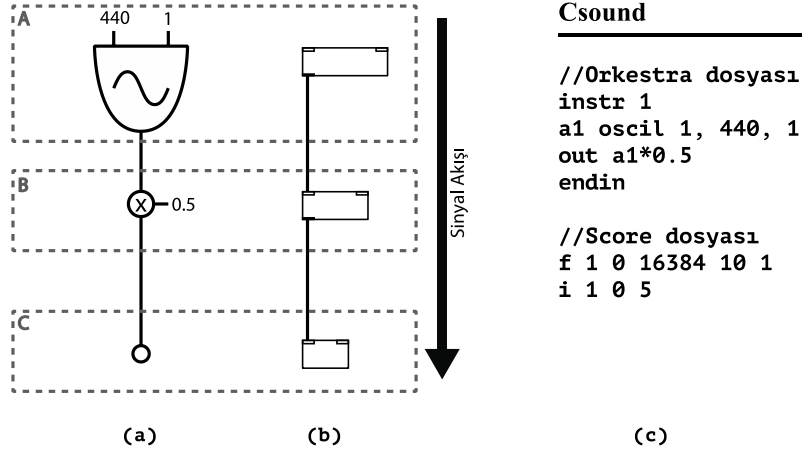


Şekil 3 - Analog modüler sentezleyicilerde ve Pure Data’da kablolama (patching)

Şekil-4’deki uygulamada frekansı 440 Hz, genliği 1 olan bir sinüs dalga osilatörünün çıkışı 0,5 ile çarpılarak yarıya düşürülmüş ve ilgili çıkışa gönderilmiştir. Şekil 4(a)’da uygulamanın grafiksel notasyonu, Şekil 4(b)’de ise Pure Data patch dosyası görülmektedir.

Uygulama A, B ve C olarak adlandırılmış üç aşamada gösterilmiştir. A bölgesinde 440 Hz frekanslı, genliği 1 olan sinüs dalga osilatörü, Pure Data’da osc~ nesnesi ile oluşturulmuştur. Osilatör frekansı ilgili nesne kutusu içerisinde osc~ nesne isminin sağ tarafında 440 olarak belirtilmiştir. Pure Data’da osc~ ile üretilen sinüs dalganın varsayılan genlik değeri 1’dir.

B bölgesinde dalga genliği 0,5 ile çarpılarak yarıya düşürülmektedir. Grafik notasyonda “x” işareti ile gösterilen çarpma işlemi Pure Data’da “*~” nesnesi ile gerçekleştirilmiştir. Bu işlemde 0,5 parametresi yine nesne kutusu içerisinde yazılmıştır. C bölgesinde ise sinyal grafiksel notasyonda küçük daire simgesiyle gösterilen çıkış noktasına gönderilmektedir. Csound ortamında aksi belirtilmedikçe varsayılan çıkış bir ses dosyasıdır. Pure Data’da ise ses sinyalleri dac~ olarak adlandırılan sayısal – analog dönüştürücüye gönderilerek gerçek zamanlı olarak işitilebilir. Burada sayısal – analog dönüştürücü bir bilgisayara bağlı olan ses donanımını ifade eder. Şekil 4(c)’de ise uygulamanın Csound kodu görülmektedir.

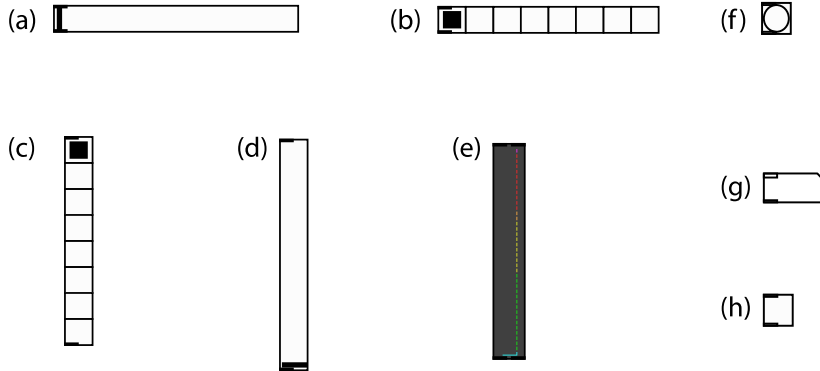


Şekil 4 - Genliği yarıya düşürülmüş bir basit sinüs osilatörün Pure Data ve Csound uygulaması

Görsellik ve Etkileşim

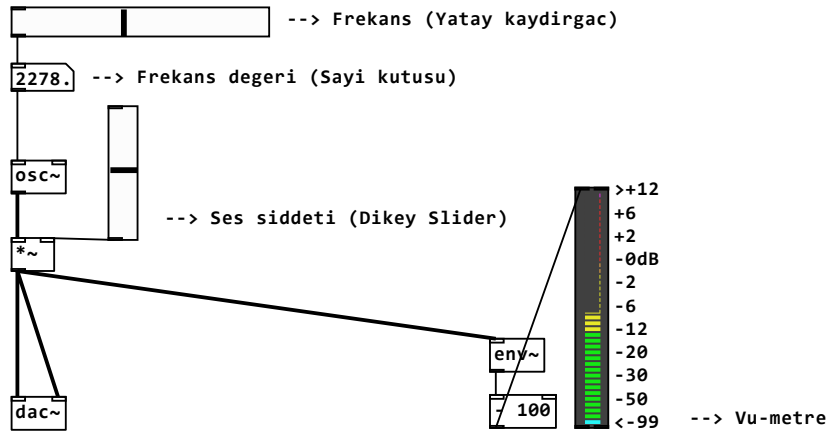
Pure Data’da patch dosyaları ile kullanıcı arasındaki etkileşim Görsel Kullanıcı Arayüzleri (GUI) adı verilen özel nesneler ile gerçekleştirilir. Çalışma sırasında kullanıcıya gerçek zamanlı etkileşim olanağı tanıyan bu nesneler veri üretmek veya görüntülemek amacıyla kullanılırlar (Farnell, 2010: 158).

Şekil 5’de görülen Pure Data’ya ait GUI’ler kullanıcının fare aracılığı ile etkileşime geçmesi sonucu sayısal değerler (mesajlar) üretirler. Üretilen bu veriler, oluşturulan bağlantılar üzerinden ilgili nesnelere ulaştırılır ve gerektiği biçimde işlenir. Tersî biçimde, GUI’lerin girişlerine ulaşan sayısal değerler ise bu nesneler tarafından görselleştirilir.



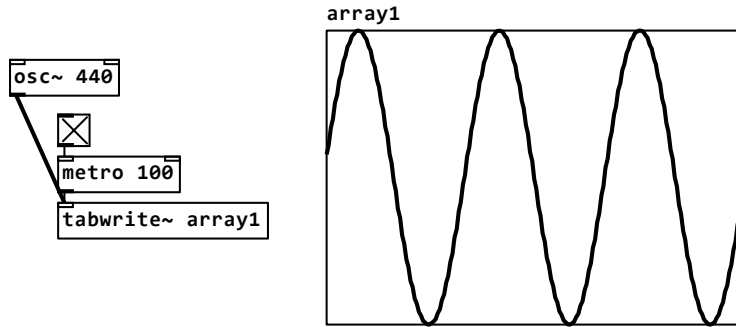
Şekil 5 - (a) Yatay kaydırgaç, (b) Yatay radyo kutusu, (c) Dikey radyo kutusu, (d) Dikey kaydırgaç, (e) Vu-Metre, (f) Bang, (g) Sayı kutusu, (h) Toggle

Şekil 6’daki örnek, genliği ve frekansı kullanıcı tarafından gerçek zamanlı olarak kontrol edilebilen bir sinüs osilatörünü göstermektedir. Frekans değeri bir yatay kaydırgaç ile değiştirilerek bu değer sayı kutusu tarafından görüntülenmektedir. Sinyal genliği de (ses şiddeti) bir dikey kaydırgaç ile belirlenmekte ve bir Vu-metre aracılığı ile dB cinsinden görüntülenmektedir.



Şekil 6 - Frekans ve genlik parametreleri Pure Data GUI'leri ile gerçek zamanlı olarak kontrol edilebilen bir sinüs osilatör uygulaması

Pure Data'da *array* nesnesi belirlenen boyutta bir tablo (dizi) oluşturarak bu tablo içeriğini noktalar veya çizgiler ile görselleştirir. Şekil 7'de bir kosinüs sinyalinin *array* nesnesi ile görselleştirildiği örnek patch dosyası görülmektedir.



Şekil 7 - 440 Hz frekanslı kosinüs dalganın array nesnesi ile görüntülenmesi

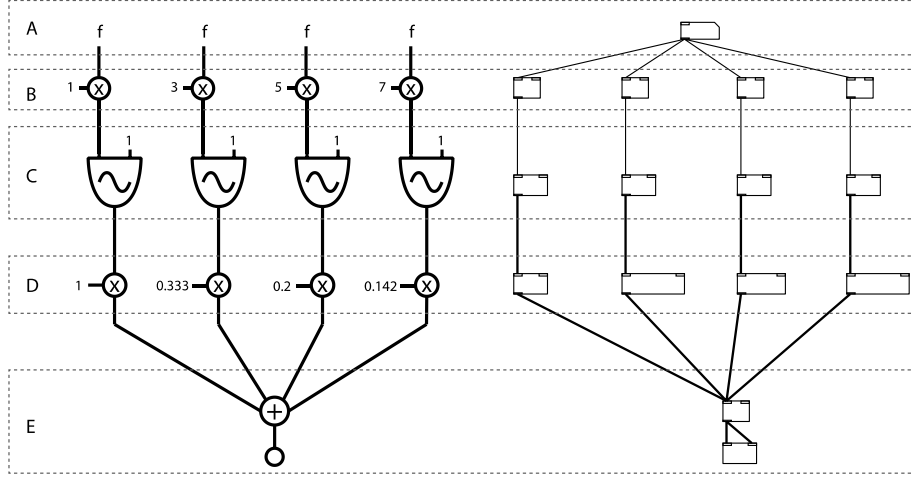
Ekleme Sentezleme Uygulaması

Fransız matematikçi Fourier 1807'de karmaşık dalga biçimlerinin genlikleri ve frekansları bilinen basit dalgaların birleşimi şeklinde ifade edilebileceğini göstermiştir (Russ, 2004: 109). Bu prensibe dayanan artırmalı sentezleme, basit dalga biçimlerini birleştirilerek daha karmaşık dalga biçimleri elde etme yöntemidir. Şekil 8'de Pure Data ortamında gerçekleştirilmiş bir artırmalı sentezleme uygulamasına ait grafik notasyon ve patch dosyası görülmektedir.

A ile gösterilen bölümde tüm osilatörlere gönderilecek temel frekans değeri Pure Data ortamında sayı nesnesi ile ifade edilmiştir. Bu nesne Pure Data'ya özgü bir GUI (Görsel Kullanıcı Arayüzü) olup, değeri çalışma sırasında gerçek zamanlı olarak değiştirilebilmektedir.

B bölümünde artırmalı sentezlemenin hangi harmonikler kullanılarak gerçekleştirileceği belirlenmiştir. Bu örnekte 1, 3, 5 ve 7'nci harmonikler kullanılmıştır. Sayı kutusundan gelen temel frekans değeri ilgili harmonik numarası ile çarpılarak her bir osilatöre gönderilmiştir. C ile gösterilen bölgede osilatörler Pure Data'ya ait *osc~* nesnesi ile oluşturulmuştur.

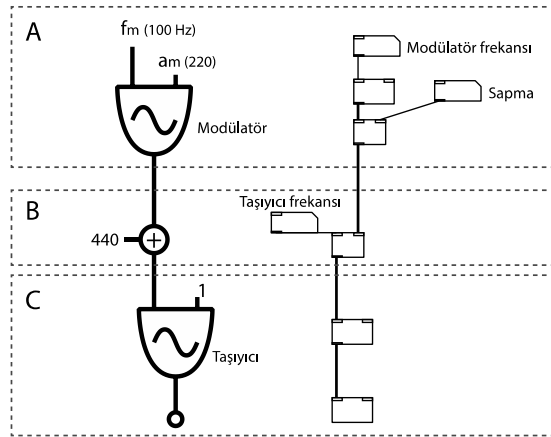
D bölümünde her bir osilatörün çıkışı oranlanarak harmoniklerin seviyeleri belirlenmiştir. Bu işlem Pure Data'nın *~ nesnesi ile gerçekleştirilmiştir. Son bölüm olan E bölümünde ise tüm osilatörlerin çıkışları toplanarak ses işlemcisine gönderilmiştir.



Şekil 8 - Eklemeli sentezleme tekniğinin grafiksel notasyon ile gösterimi (solda) ve Pure Data uygulaması (sağda)

Frekans Modülasyonu ile Sentezleme Uygulaması

FM sentezleme tekniği John Chowning tarafından 1960'ların sonlarına doğru geliştirilmiştir. FM radyo iletişimi ile aynı prensiple çalışan bu teknikte taşıyıcı olarak adlandırılan bir osilatörün frekansı, modülatör adı verilen başka bir osilatör tarafından kontrol edilerek tınısal bakımdan daha zengin sesler elde edilmesi hedeflenir (Miranda, 2002: 23-24).



Şekil 9 - FM sentezleme tekniğinin grafiksel notasyon ile gösterimi ve Pure Data uygulaması

Şekil 9'daki örnekte Pure Data ile bir FM sentezleme uygulaması görülmektedir. A bölümünde modülatör osilatörü osc~ nesnesi ile oluşturulmuştur. Pure Data'da osc~ nesnesi -1 ile +1 arasında değerler üretir. Bu aralık FM sentezlemede sapma (deviation) olarak adlandırılan

modülatör genliği için çok düşük olduğundan osc~ nesnesinin çıkışı bir sayı kutusu ile belirlenen sapma değeri ile çarpılmaktadır.

B bölümünde modülatörün çıkışı bir sayı kutusu ile belirlenen taşıyıcı frekansı ile toplanmaktadır. Bu aşama modülasyon olayının gerçekleştiği aşamadır. C’de ise genliği 1 olan taşıyıcı osilatör başka bir osc~ nesnesi ile oluşturularak çıkışı ses donanımına gönderilmiştir.

SONUÇ

Temel ses sentezleme tekniklerine yönelik teorik bilginin öğretilmesinde sinyal akışının ve donanımlar arası iç bağlantıların görselleştirilmesi amacıyla grafiksel notasyondan faydalanılabilir. Ancak grafiksel notasyon, sentezleme sürecinin ürünü olan ses sinyalinin tınısal özelliği hakkında herhangi bir fikir vermemektedir. Sesin işitilebilmesi ve böylelikle tınısal etkinin algılanması ancak ilgili sentezleme tekniğinin sentezleyici araçlar üzerinde uygulanması ile mümkün olabilir. İç bağlantıları sabit olan performans tipi sentezleyiciler yeterince esnek yapıda olmadıklarından her sentezleme tekniğine uygun bir sentezleyicinin ayrıca temin edilmesi gerekmektedir. Alternatif olarak daha esnek yapıdaki modüler sentezleyicilerden de faydalanılabilir. Ancak her iki çözüm de belirli bir maddi yükümlülüğü bereberinde getirmektedir.

Pure Data, eğitim sürecinde donanımsal sentezleyicilere olan ihtiyacı ortadan kaldıracak alternatif bir çözüm olarak karşımıza çıkmaktadır. Yazılımın ücretsiz olarak temin edinebilmesi, kullanıcıya herhangi bir lisans kısıtlaması getirmemesi ve Linux, Mac Os X ile Windows işletim sistemleri üzerinde çalışabilmesi, özellikle ses sentezleme teknikleri eğitimi verilecek dersliklerin oluşturulması aşamasında oldukça önemli kazanımlar sağlayabilir.

Kod tabanlı yazılımlar ile karşılaştırıldığında Pure Data’nın grafiksel tasarım arayüzünün sentezleme tekniklerinin gösteriminde geleneksel bir yöntem olan grafiksel notasyona paralellikler gösterdiği göze çarpmaktadır. Örnek uygulamalarda da görüldüğü üzere Pure Data’nın bu özelliğinin teorinin pratiğe uyarlanması sürecinde oldukça önemli kolaylıklar sağlayacağı, özellikle tasarım aşamasında soyut kavramları somutlaştırarak daha kolay anlaşılabilir hale getireceği söylenebilir.

Pure Data’da görsel kullanıcı arayüzleri (GUI) olarak adlandırılan özel nesneler ile parametrelerin gerçek zamanlı olarak değiştirilebilmesi ve sonucun doğrudan ses donanımına gönderilerek işitilebilir hale gelmesi, anlık parametre değişimlerinin sentezleme sürecine etkisinin doğrudan gözlemlenebilmesi açısından önemli bir kazanım olarak değerlendirilebilir.

KAYNAKLAR

- CHUNG, B. WC. (2013). *Multimedia Programing With Pure Data*. UK: Packt Publishing.
- FARNELL, A. (2010). *Designing sound*. The MIT Press.
- MIRANDA, E. R. (2002). *Computer Sound Design: Synthesis Techniques and Programming*. Focal Press.
- ÖNEN, U. & Pasinlioğlu, T. (2011). *Synthesizer Teknolojileri ve Programlama*. İstanbul: Çitlembik Yayınları.
- ROADS, C. (1996). *The Computer Music Tutorial*. The MIT Press.
- RUSS, M. (2004). *Sound Synthesis and Sampling*. Focal Press.